DOCUMENT RESUME

ED 350 999 IR 015 814

AUTHOR Pyzdrowski, Anthony S.; DeNardo, Anette M.
TITLE Creating a Computer Simulator Package for a Hypothetical Computer Architecture.
PUB DATE Mar 92
NOTE 7p.; Paper presented at the Annual Conference of the Eastern Educational Research Association (15th, Hilton Head, SC, March 5, 1992).
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Authoring Aids (Programing); *Computer Assisted Instruction; *Computer Science Education; *Computer Simulation; Computer System Design; Higher Education; Hypermedia; Instructional Effectiveness; Microcomputers; Programing; Student Attitudes
IDENTIFIERS *Computer Architecture

ABSTRACT

This paper describes three computer-based simulators developed to support an undergraduate computer architecture course. Using a hypermedia authoring package, a simulator was created for each of the three course content areas: (1) introduction of the hypothetical computer structure--contains a tutorial, a quiz, and a simple program that presents background information and introduces the architecture of the hypothetical machine and the first 11 fundamental machine instructions; (2) presentation of the microsequence operations--includes a quiz section which tests student understanding of concepts; and (3) actual programming of the hypothetical computer--allows students to write and execute programs using a graphical presentation of the program. Results from research on the class indicated that students' responses were favorable to the incorporation of the simulators in the class; students showed lower levels of frustration and a higher level of understanding of abstract materials; students shifted from lecture and classroom questioning on specific items as a means of obtaining information to the consultation of the appropriate simulator to obtain the answer; and the class showed significant improvements. Two figures illustrate sample screens from Simulators One and Two. (ALF)

Creating a Computer Simulator Package for a Hypothetical Computer Architecture

Anthony S. Pyzdrowski
Department of Mathematics and Computer Science
California University of Pennsylvania
California, PA 15419

Anette M. DeNardo
Department of Curriculum and Instruction
West Virginia University
Morgantown, WV 26505

ABSTRACT

This paper describes three computer-based simulators developed to support an undergraduate computer architecture course. Two simulators were written using a hypermedia authoring package. The first contains a tutorial, quiz, and simple program simulator. The second demonstrates data flow for each microsequence of the instruction set at the machine level. Both simulators present the material in a graphical environment.

The third simulator was written in C and allows the students to write programs for the hypothetical computer. The students are able to print a core dump of the fetch and execution cycles for each instruction of the program. In this way they can closely examine the contents of all CPU components during program execution.

INTRODUCTION

Computer-based simulators are programs that model realistic situations because they require active participation by the users in initiating and performing inquiries, decisions, and actions. In addition to providing an opportunity to acquire skills, attain new concepts, and engage in problem solving, simulations can provide students with immediate feedback and rapid repeatability without concern for the time and expense involved in using real materials (Gorrell, Cuevas, & Downing, 1988; Lunetta & Hofstein, 1981; McGuire, 1976; Spain, 1984; Strickland & Poe, 1989). Most educational settings require simple forms of simulators. Details are changed or omitted to provide thought or engage the student in participation (Strickland & Poe). Therefore, most simulators possess three attributes: (a) they represent a real situation, (b) they provide the user with certain controls over the situation, and (c) they omit irrelevant or unimportant variables (Gagné, 1962).

A variety of educational settings use computer-based simulations. The study of computer architecture is one such area. Because students cannot study the internal architecture of a computer system, computer programs to simulate the architecture are frequently used. Unfortunately, research on the effectiveness of computer simulations in this area is almost nonexistent.

This software development was motivated by the need for supportive computer learning aids to supplement a university-level computer architecture course. The content of the computer architecture course is not typical of the courses encountered by computer science students. This situation makes for difficulty in learning. Reigeluth and Schwartz (1989) contend that "computer-based simulations can provide efficient,

effective, and highly motivational instruction that can readily serve the need for individualization." Reigeluth and Schwartz (1989) also state that simulations "enhance the transfer of learning by teaching complex tasks in an environment that approximates the real world setting in certain ways." Boblick (1972) and Cavin and Lagowski (1978) found that students who used simulators for physical experiments performed as well as or better than students who actually performed the experiments.

Therefore, three simulators were written to support a college-level course in computer architecture. This paper presents an overview of the simulators.

OBJECTIVE

The objective of the project is to provide support computer packages for a computer architecture course. The material covered in the computer architecture course is diverse and requires a variety of specialized support simulators. The first content area involves the introduction of the hypothetical computer structure. This section presents the general components of the internal structure and the operation of the computer on a macro level. A firm understanding of the concepts and components is required prior to the introduction and use of the machine level instruction set. Not only is the presentation of the instruction set important but also the presentation of program examples helps the student understand the operation of the computer. The presentation of program executions should emphasize the data transfers which occur within the computer. Understanding the transfer of data helps the student make the transition of a macro level understanding of the structure to the required micro level understanding of the structure. The next content area pertains to the presentation of the microsequence operations that each machine instruction process in order to have the computer perform the instruction. The microsequences control the transfer of data within the computer structure. The last content area is the actual programming of the hypothetical computer structure. A means for the student to write programs using the machines instruction set and execute the programs on the computer, if it existed. The execution of the programs should provide the student with the contents of each component of the computer after each instruction is executed.

TECHNIQUE

A simulator was created for each of the three course areas. The three computer simulators of the hypothetical computer architecture are presented differently. Based on the content of the package and its desired outcome, each package is presented in a way that makes it appealing to the student.

The first simulator contains three parts all written using *ToolBook* by *Asymetrix*. The first section presents background information, introduces the architecture of the hypothetical machine, and presents the first 11 fundamental machine instructions. This is the first encounter the student has with the computer packages. Presenting the introduction in a colorful and graphical self-paced tutorial with numerous "hotwords" for additional explanation maintains the interest of the students without inhibiting them from exploring the material to their satisfaction.

The second section of the simulator includes a quiz section which tests student understanding of those concepts discussed in the tutorial. Students are encouraged to achieve at least an 80 percent proficiency on each of the three parts of the quiz before proceeding on to the simulator. The three quiz sections are graphically colorful with audio support. The quiz sections create an adventure environment that rewards the student for a successful competency achievement.

Part three of the first simulator demonstrates the operation of the hypothetical computer through a graphical presentation of program execution. A graphical presentation of the physical components--registers and memory--of the hypothetical computer appear on the display (see Figure 1). Several example programs are available for execution by the student. These example programs enable the student to observe the changes in the computer during program execution. When executing the instructions of a program, the simulator changes the contents of the displayed components. Also included with the components is an outline of the execution cycle with the current step highlighted. By selecting a designated button, the student initiates the execution of each instruction. This student-invoked stepping of the program allows a self-paced examination of the events related to each instruction. Once the students have an understanding of the instructions and their operations, they can enter their own programs. The student writes these programs in binary or octal machine code. This feature allows the students to extend their understanding of the computer's instruction

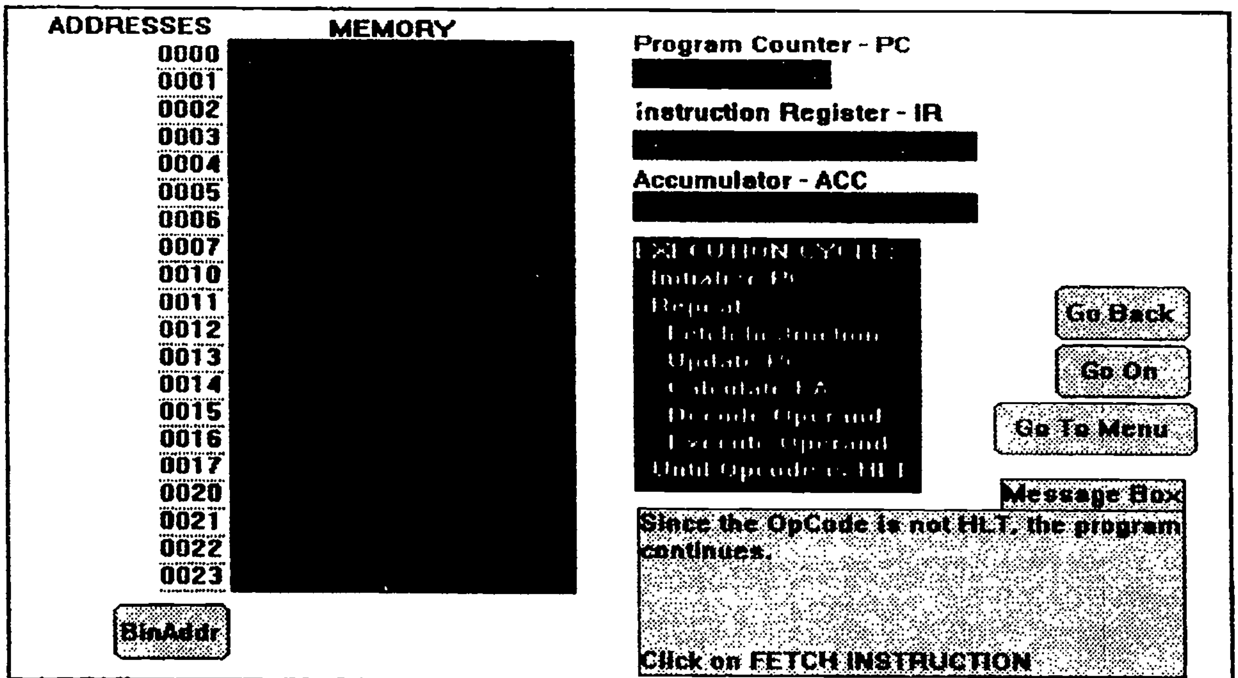

Figure 1 Sample Screen from Simulator One

set and refine their program writing skills for the new machine language. The student can review the operations of the computer until a thorough understanding is obtained.

The second hypermedia simulator presents definitions of old and new components and the complete graphical rendering of the computer. This graphical representation of the structure contains buttons. When the mouse is pointed to these buttons and clicked, further explanations for each component are provided. These components include the ALU (arithmetic logic unit) and flags, accumulator, index registers, control unit, IR (instruction register), PC (program counter), memory, memory address register, memory buffer register, and all busses that interconnect the components.

Following the graphical presentation of the computer, the simulator displays a menu of 26 instructions. From this menu the student can select a speed of presentation and an instruction of interest. Once a student selects an instruction, the simulator re-displays the structure with the microsequences of that instruction. The student selects one microsequence at a time. Upon selection of a microsequence, the simulator, using colors, traces the flow of data throughout the structure (see Figure 2). These colored paths flow throughout the structure at a rate that proportionally reflects the real operation time. The colored trace remains on the display for student scrutiny. The structure is cleared of the coloring upon selection of the next microsequence. All microsequences are repeatable. When the entire sequence necessary for the implementation of the selected instruction is complete, the simulator returns to the menu.

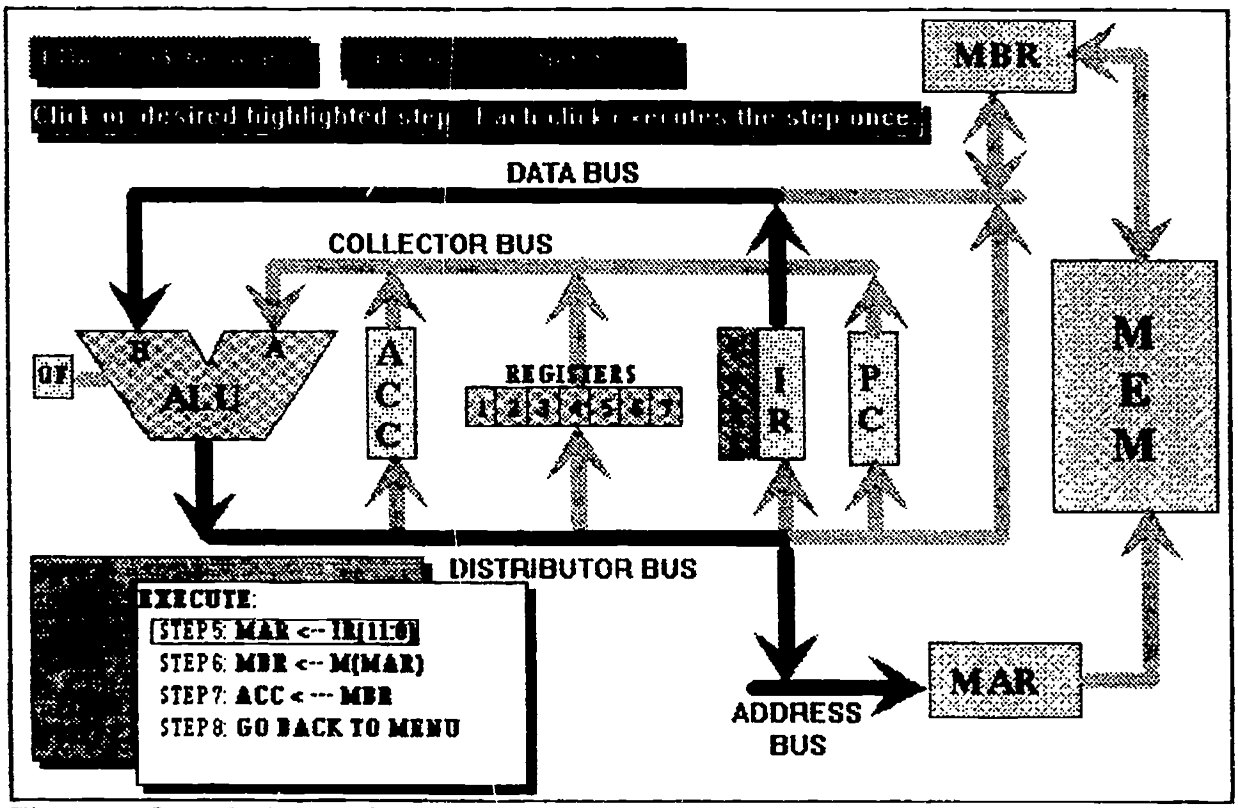

Figure 2 Sample Screen from Simulator Two

Near the end of the semester, the students use a simulator containing the entire instruction set of the hypothetical computer. This simulator accepts programs written by the students and demonstrates the execution sequence for the program.

The final simulator is written in C. This simulator implements all instructions of the hypothetical computer. Unlike the previous simulators, this package does not provide the student with any educational instruction. Its sole purpose is to allow the student to write programs for this hypothetical computer and observe their execution. The student enters a program from either the keyboard or a file. Entering a program from a file, the student can write larger and more involved programs while allowing for changes. The output of the simulator can be sent to the display or another output device. Alternate output devices allow for a permanent copy of the program sequence so that the student can better study and understand the program execution. The output contains the contents of all registers for each instruction executed until the program terminates. Additionally, the contents of memory can be output, or core dumped, for examination. There is no error detection or help within this simulator; its task is to test the effectiveness of the students' programming abilities of the new machine presented in the previous two learning simulators.

CONCLUSIONS

Student responses were favorable toward the incorporation of the simulators in the class. After the introduction of the packages in the class, students showed lower levels of frustration pertaining to the abstract material and a higher level of understanding and comprehension of the material. Students shifted from lecture and classroom questioning on specific items as a means of obtaining information to the consultation of the appropriate simulator to obtain the answer. Results from research on the class showed significant student improvements (DeNardo & Pyzdrowski, in press).

REFERENCES

Boblick, J. M. (1972). Discovering the conservation of momentum through the use of a computer simulation of a one-dimensional elastic collision. Science Education, 56(3), 337-344.

Cavin, C. S., & Lagowski, J. J. (1978). Effects of computer simulated or laboratory experiments and student aptitude on achievement and time in college general chemistry laboratory course. Journal of Research in Science Teaching, 15(6), 455-463.

Gagné, R. M. (1962). Simulators. In R. Glaser (Ed.), Training research and education (pp. 223-246). Pittsburgh: University of Pittsburgh Press.

Gorrell, J., Cuevas, A., & Downing, H. (1988). Computer simulations of classroom behavior problems. Computers and Education, 12(2), 283-287.

Lunetta, V. N., & Hofstein, A. (1981). Simulations in science education. Science Education, 65(3), 243-252.

McGuire, C. (1976). Simulation technique in the teaching and testing of problem-solving skills. Journal of Research in Science Teaching, 13(2), 89-100.

Reigeluth, C. M., & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. Journal of Computer-Based Instruction, 16(1), 1-10.

Spain, J. D. (1984). Computer center: It's time to take inventory. The American Biology Teacher, 46(6), 347-349.

Strickland, R. M., & Poe, S. E. (1989). Developing a CAI graphics simulation model: Guidelines. Technological Horizons in Education Journal, 16(7), 88-92.